

Amortized Averaging Algorithms for Approximate Consensus

Bernadette Charron-Bost¹Matthias Függer²Thomas Nowak³¹ CNRS, École polytechnique² Max-Planck-Institut für Informatik³ Université Paris-Sud

Abstract

We introduce a new class of distributed algorithms for the approximate consensus problem in dynamic rooted networks, which we call *amortized averaging algorithms*. They are deduced from ordinary averaging algorithms by adding a value-gathering phase before each value update. This allows their decision time to drop from being exponential in the number n of processes to being linear under the assumption that each process knows n . In particular, the *amortized midpoint algorithm*, which achieves a linear decision time, works in completely anonymous dynamic rooted networks where processes can exchange and store continuous values, and under the assumption that the number of processes is known to all processes. We then study the way amortized averaging algorithms degrade when communication graphs are from time to time non rooted, or with a wrong estimate of the number of processes. Finally, we analyze the amortized midpoint algorithm under the additional constraint that processes can only store and send quantized values, and get as a corollary that the 2-set consensus problem is solvable in linear time in any rooted dynamic network model when allowing all decision values to be in the range of initial values.

1 Introduction

This paper studies the problem of *approximate consensus*, i.e., the task for a set of processes, each of them with an initial real scalar value, to decide on values that lie in an ε -neighborhood of each other and in the range of the initial values. It has many applications, e.g., for geometric coordination tasks or clock synchronization, which often have to be solved in mobile ad-hoc networks under quite adverse constraints.

In recent work [5], we proved that approximate consensus is solvable in a dynamic network model if and only if all occurring communication graphs are rooted, i.e., contain a rooted spanning tree. This spanning tree, as well as the root can change completely from one communication graph to the next. There is hence no stability condition necessary for solving approximate consensus. In fact, we showed that it suffices to restrict our attention to the simple class of *averaging algorithms* to solve approximate consensus whenever it is indeed solvable. In these algorithms, processes have a single scalar state variable, which they repeatedly update to a weighted average of the values they received from their neighbors. We proved that, if ϱ is a lower bound on the weights used in the averaging steps of an averaging algorithm, then it achieves ε -agreement in $O(n\varrho^{-n} \log \frac{1}{\varepsilon})$ rounds where n is the number of processes. Moreover, example graphs exist showing that averaging algorithms can really need an exponential time until they achieve ε -agreement.

We now list the main contributions of the present paper:

1. We propose a new analysis technique of averaging algorithms based on the notion of ϱ -*safeness*, introduced in [16, 9], which is a generalization of the lower bound condition on the parameter ϱ discussed above. This technique focuses on the essential properties needed for contracting the range of current values in the system by directly focusing on the set of values and not on the weights used in the averaging steps, as done classically. Together with a graph-theoretic reduction that already played a key role in [5], it enables short proofs of the convergence and of upper bounds on the contraction rate of (existing and new) averaging algorithms.

2. We introduce the notion of *amortization* of averaging algorithms, which consists in inserting a value-gathering phase before each averaging step. This additional phase, surprisingly, transforms averaging algorithms into “turbo versions” of themselves and takes their decision time from being exponential [5] to being polynomial in the number of processes.
3. We combine these two ideas in the design of the *amortized midpoint algorithm*, which we prove to have an optimal contraction rate and whose decision time is linear in the number of processes, which is also optimal. More specifically, it achieves ε -agreement in $O(n \log \frac{1}{\varepsilon})$ rounds where n is the number of processes. This algorithm neither relies on any stability of the communication topology nor on any way of distinguishing two processes. It thus works in completely dynamic anonymous networks.

The well-functioning of the amortized midpoint algorithm does, however, rely on two fundamental hypotheses: knowing the number of processes and the rootedness of all occurring communication graphs. However we demonstrate that, even with an erroneous estimate on the number of processes or when communication graphs sometimes fail to be rooted, the amortized midpoint algorithm still achieves ε -agreement, albeit at a later time. In fact, this graceful degradation property holds for all amortized averaging algorithms, which shows that a certain part of the often observed robustness of averaging algorithms actually carries over to their amortized versions.

The linear convergence time of the amortized midpoint algorithm, combined with its great versatility and robustness, is especially striking, notably when comparing it to classical averaging algorithms such as the *equal-neighbor* algorithm. It is well known that the latter averaging algorithms may give an exponential decision time in the number of processes.

One of the first steps to go from exponential to polynomial decision time was done by Olshevsky and Tsitsiklis [19] who presented a cubic-time averaging algorithm with time-varying bidirectional topologies. This result was later improved to quadratic time [17] when the update rules at every time correspond to a doubly stochastic matrix. Very recently, Olshevsky presented a linear-time algorithm in time-constant bidirectional communication graphs [18]. This result was preceded by other attempts at lowering the convergence time in several other special cases (e.g., [11, 13, 1, 23]). The sum of these efforts makes the time-linearity of the relatively simple amortized midpoint algorithm in arbitrarily dynamic directed anonymous networks all the more so striking.

The robustness against changes in the hypotheses goes even further: If processes do not dispose of real-valued variables, but only variables of finite precision, then the amortized midpoint algorithm achieves ε -consensus in linear time as long as it is not obviously impossible, i.e., whenever ε is larger than the variables’ precision. If ε is smaller than the precision, then ε -agreement cannot be achieved in all rooted network models because of the impossibility of exact consensus [21]. In fact, this property of achieving ε -agreement whenever it is possible is specific to the amortized midpoint algorithm and does not hold for general amortized averaging algorithms and even less so for non-amortized averaging algorithms.

By choosing ε equal to the precision, we get as a corollary that the *2-set consensus problem* is solvable in linear time in all rooted dynamic network models. This new result on k -set consensus holds when all decision values are only constrained to be in the range of initial values (instead of being equal to initial values), which sheds new light on the role of the validity condition in the specification of this classical problem in distributed computing.

While the practical implications of having a fast approximate consensus algorithm are immediately visible in man-made communication networks, our results go even beyond that and are interesting also from a different, more analytic, perspective. Many natural phenomena, like bird flocking [8], synchronization of coupled oscillators [22], opinion dynamics [12], or firefly synchronization [15], can be modeled and explained via agents that execute averaging algorithms. However, there is an important mismatch between the fast convergence times observed in nature and the theoretical worst-case times, which often are exponential or worse [8]. In this sense, our results provide a step towards closing this gap by suggesting that the agents in these natural systems in fact operate on a different, slower, time scale than their environments, and that this inertia actually contributes positively to faster convergence.

The rest of the paper is organized as follows: Section 2 gives the problem and model definition, as well as introducing the concept of averaging algorithms. Section 3 introduces the analysis technique we use and gives tight upper bounds on the contracting rate of averaging algorithms. The concept of amortization of averaging algorithms is given in Section 4. Several amortized versions of averaging

algorithms are introduced and studied, in particular the amortized midpoint algorithm. Resiliency and robustness of these algorithms is studied in Section 5. Finally, Section 6 extends the analysis to variables of finite precision.

2 Approximate consensus and averaging algorithms

We assume a distributed, round-based computational model in the spirit of the Heard-Of model by Charron-Bost and Schiper [6]. A system consists in a set of processes $[n] = \{1, \dots, n\}$. Computation proceeds in *rounds*: In a round, each process sends its state to its outgoing neighbors, receives values from its incoming neighbors, and finally updates its state based. The value of the updated state is determined by a deterministic algorithm, i.e., a transition function that maps the values in the incoming messages to a new state value. Rounds are communication closed in the sense that no process receives values in round k that are sent in a round different from k .

Communications that occur in a round are modeled by a directed graph $G = ([n], E(G))$ with a self-loop at each node. The latter requirement is quite natural as a process can obviously communicate with itself instantaneously. Such a directed graph is called a *communication graph*. We denote by $\text{In}_p(G)$ the set of incoming neighbors of p and by $\text{Out}_p(G)$ the set of outgoing neighbors of p in G .

The *product* of two communication graphs G and H , denoted $G \circ H$, is the directed graph with an edge from (p, q) if there exists $r \in [n]$ such that $(p, r) \in E(G)$ and $(r, q) \in E(H)$. We easily see that $G \circ H$ is a communication graph and, because of the self-loops, $E(G) \cup E(H) \subseteq E(G \circ H)$.

A *communication pattern* is a sequence $(G(k))_{k \geq 1}$ of communication graphs. For a given communication pattern, $\text{In}_p(k)$ and $\text{Out}_p(k)$ stand for $\text{In}_p(G(k))$ and $\text{Out}_p(G(k))$, respectively.

Each process p has a *local state* s_p the values of which at the end of round $k \geq 1$ is denoted by $s_p(k)$. Process p 's initial state, i.e., its state at the beginning of round 1, is denoted by $s_p(0)$. Let the *global state* at the end of round k be the collection $s(k) = (s_p(k))_{p \in [n]}$. The *execution* of an algorithm from global initial state $s(0)$, with communication pattern $(G(k))_{k \geq 1}$ is the unique sequence $(s(k))_{k \geq 0}$ of global states defined as follows: for each round $k \geq 1$, process p sends $s_p(k-1)$ to all the processes in $\text{Out}_p(k)$, receives $s_q(k-1)$ from each process q in $\text{In}_p(k)$, and computes $s_p(k)$ from the incoming messages, according to the algorithm's transition function.

When the structure of states allows each process to record and to relay information it has received during any period of K rounds for some positive integer K , we may be led to modify time-scale and to consider blocks of K consecutive rounds, called *macro-rounds*: macro-round ℓ is the sequence of rounds $(\ell-1)K+1, \dots, \ell K$ and the corresponding information flow graph, called *communication graph at macro-round ℓ* , is the product of the communication graphs $G((\ell-1)K+1) \circ \dots \circ G(\ell K)$.

2.1 Consensus and approximate consensus

A crucial problem in distributed systems is to achieve agreement among local process states from arbitrary initial local states. It is a well-known fact that this goal is not easily achievable in the context of dynamic network changes [21, 5], and restrictions on communication patterns are required for that. We define a *network model* as a non-empty set \mathcal{N} of communication graphs, those that may occur in communication patterns.

We now consider the above round-based algorithms in which the local state of process p contains two variables x_p and dec_p . Initially the range of x_p is $[0, 1]$ and $\text{dec}_p = \perp$ (which informally means that p has not decided yet).⁴ Process p is allowed to set dec_p to the current value of x_p , and so to a value v different from \perp , only once; in that case we say that p *decides* v .

For any $\varepsilon \geq 0$, an algorithm *achieves ε -agreement with the communication pattern $(G(k))_{k \geq 1}$* if each execution from a global initial state as specified above and with the communication pattern $(G(k))_{k \geq 1}$ fulfills the following three conditions:

ε -Agreement. The decision values of any two processes are within ε .

Validity. All decided values are in the range of the initial values of processes.

Termination. All processes eventually decide.

⁴In the case of *binary consensus*, x_p is restricted to be initially from $\{0, 1\}$.

An algorithm *solves approximate consensus* in a network model \mathcal{N} if for any $\varepsilon > 0$, it achieves ε -agreement with each communication pattern formed with graphs all in \mathcal{N} .

In a previous paper [5], we proved the following characterization of network models in which approximate consensus is solvable:

THEOREM 1 ([5]). *The approximate consensus problem is solvable in a network model \mathcal{N} if and only if each graph in \mathcal{N} has a rooted spanning tree.*

2.2 Averaging algorithms

We focus on *averaging algorithms* in which at each round k , every process p updates x_p to some weighted average of the values it has just received: formally, if $m_p(k-1)$ is the minimum of the values $\{x_q(k-1) \mid q \in \text{In}_p(k)\}$ received by p in round k and $M_p(k-1)$ is its maximum, then

$$m_p(k-1) \leq x_p(k) \leq M_p(k-1).$$

In other words, at each round k , every process adopts a new value within the interval formed by the values of its incoming neighbors in the communication graph $G(k)$.

Since we strive for distributed implementations of averaging algorithms, weights in the average update rule of process p are required to be locally computable by p . They may depend only on the set of values received by p at round k , as is the case, for instance, with the update rule of the *mean-value algorithm*:

$$x_p(k) = \frac{1}{|V_p(k)|} \sum_{v \in V_p(k)} v \quad (1)$$

where $V_p(k) = \{x_q(k-1) \mid q \in \text{In}_p(k)\}$. In contrast, even in anonymous networks, weights in the update rule for p may depend on the *multiset* of values received by p at round k counted with their multiplicities: as an example, p 's update rule in the *equal-neighbor algorithm* is given by:

$$x_p(k) = \frac{1}{|\text{In}_p(k)|} \sum_{q \in \text{In}_p(k)} x_q(k-1). \quad (2)$$

Observe that the decision rule is not specified in the above definition of averaging algorithms: the decision time immediately follows from the number of rounds that is proven to be sufficient to reach ε -agreement.

Let $\varrho \in]0, 1/2]$; an averaging algorithm is ϱ -safe in \mathcal{N} if at any round k of each of its executions with communication patterns in \mathcal{N} , each process adopts a new value within the interval formed by its neighbors in $G(k)$ not too close to the boundary:

$$\varrho M_p(k-1) + (1-\varrho)m_p(k-1) \leq x_p(k) \leq (1-\varrho)M_p(k-1) + \varrho m_p(k-1).$$

Clearly if an averaging algorithm is ϱ -safe, then it is ϱ' -safe for any $\varrho' \leq \varrho$. We also easily check the following property of the equal-neighbor and mean-value algorithms.

PROPOSITION 2. *In any network model with n processes, the equal-neighbor algorithm and the mean-value algorithm are both $(1/n)$ -safe.*

Finally, an averaging algorithm is α -contracting in \mathcal{N} if at each round k of each of its executions with communication patterns in \mathcal{N} , we have

$$\delta(x(k)) \leq \alpha \delta(x(k-1))$$

where δ is the seminorm on \mathbb{R}^n defined by $\delta(x) = \max_p(x_p) - \min_p(x_p)$.

PROPOSITION 3. *In any network model, every α -contracting averaging algorithm with $\alpha \in [0, 1[$ solves approximate consensus and achieves ε -agreement in $\lceil \log_{\frac{1}{\alpha}} \left(\frac{1}{\varepsilon} \right) \rceil$ rounds.*

Proof. Let $m(k)$ and $M(k)$ denote the minimum value and the maximum value in the network at round k , respectively; hence $\delta(x(k)) = M(k) - m(k)$.

An easy inductive argument shows that the sequences $(m(k))_{k \geq 1}$ and $(M(k))_{k \geq 1}$ are non-decreasing and non-increasing, respectively. Because the averaging algorithm is α -contracting, we have

$$\delta(x(k)) \leq \delta(x(0))\alpha^k.$$

Since $\alpha < 1$, the two sequences $(m(k))_{k \geq 0}$ and $(M(k))_{k \geq 0}$ converge to the same value. The convergence proof of each sequence $(x_p(k))_{k \geq 1}$ to a common value in the range of the initial values rests on the inequalities

$$m(0) \leq m(k) \leq x_p(k) \leq M(k) \leq M(0).$$

We have $\delta(x(k)) \leq \alpha^k$ for every integer $k \geq 0$. Plugging in $K = \lceil \log_{\frac{1}{\alpha}} \left(\frac{1}{\varepsilon} \right) \rceil$ gives $\delta(x(K)) \leq \varepsilon$. \square

3 Averaging algorithms, nonsplitness, and contraction rates

In a previous paper [5], we proved solvability of approximate consensus in rooted network models by a reduction to *nonsplit* network models: a directed graph is *nonsplit* if any two nodes have a common incoming neighbor. Indeed we showed the following general proposition:

PROPOSITION 4 ([5]). *Every product of $n - 1$ rooted graphs with n nodes and self-loops at all nodes is nonsplit.*

The main point of nonsplitness then lies in the following result:

THEOREM 5. *In a nonsplit network model, a ϱ -safe averaging algorithm is $(1 - \varrho)$ -contracting. Thus it solves approximate consensus and achieves ε -agreement in $\lceil \log_{\frac{1}{1-\varrho}} \left(\frac{1}{\varepsilon} \right) \rceil$ rounds.*

Proof. Since the communication graph $G(k)$ is nonsplit, any two processes p and q receive at least one common value v . It follows that:

$$\varrho v + (1 - \varrho)m_p(k - 1) \leq x_p(k) \leq (1 - \varrho)M_p(k - 1) + \varrho v$$

and

$$\varrho v + (1 - \varrho)m_q(k - 1) \leq x_q(k) \leq (1 - \varrho)M_q(k - 1) + \varrho v.$$

Hence

$$|x_p(k) - x_q(k)| \leq (1 - \varrho) \cdot \max_{p,q} (M_p(k - 1) - m_q(k - 1)) = (1 - \varrho) \cdot \delta(x(k - 1)).$$

This shows that the algorithm is $(1 - \varrho)$ -contracting, and the rest of the theorem follows from Proposition 3. \square

Combined with Proposition 2, we obtain an elementary proof of the classical result [3, 4] that approximate consensus is solvable in rooted network models, which does not make use anymore of Dobrushin's coefficients of scrambling matrices.

As an immediate consequence of Theorem 5, we deduce that any ϱ -safe algorithm is at best $(1/2)$ -contracting since by definition the safety coefficient ϱ cannot be larger than $1/2$. Indeed, in Section 4 we shall present an averaging algorithm that is $(1/2)$ -safe in any nonsplit network model. But do there exist other averaging algorithms with a contraction rate less than $1/2$? In the rest of the section, we show that the answer is no in the network model of nonsplit communication graphs, except in the case of two processes for which we prove that the best contraction rate is $1/3$.

THEOREM 6. *In the network model of nonsplit communication graphs with n processes, there is no averaging algorithm that is α -contracting for any $\alpha > 1/3$ if $n = 2$ and for any $\alpha > 1/2$ if $n \geq 3$.*

Proof. Let \mathcal{A} be an averaging algorithm that solves approximate consensus and that is α -contracting.

If G and H are two communication graphs with the same set of nodes $[n]$, we say that G and H are *equivalent with respect to $p \in [n]$* , denoted $G \sim_p H$, when $\text{In}_p(G) = \text{In}_p(H)$.

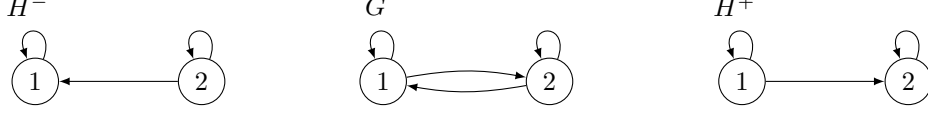


Figure 1: Communication graphs G , H^+ , and H^- used in the proof of Theorem 6 for $n = 2$

We first study the case $n = 2$ and we consider the three rounds starting with $x_1(0) = 0$, $x_2(0) = 1$, and the three communication graphs G , H^+ , and H^- (see Figure 1).

Because of the definition of an averaging algorithm, with the communication graph H^+ we have

$$x_1(1) = x_1(0) = 0 \quad \text{and} \quad x_2(1) = b \in [0, 1].$$

Likewise, with H^- , we have

$$x_2(1) = x_2(0) = 1 \quad \text{and} \quad x_1(1) = a \in [0, 1].$$

Since $G \sim_2 H^+$ and $G \sim_1 H^-$, the vector $x(1)$ obtained with the communication graph G satisfies

$$x_1(1) = a \quad \text{and} \quad x_2(1) = b.$$

Because \mathcal{A} is α -contracting by assumption, we get

$$b \leq \alpha, \quad 1 - a \leq \alpha, \quad \text{and} \quad |a - b| \leq \alpha.$$

Summing these three inequalities gives $\alpha \geq 1/3$ for $n = 2$ as required.

Assume now that $n \geq 3$. Let us consider the following two communication graphs depicted in Figure 2:

- Graph K : nodes $2, 3, \dots, n$ are fully connected and there is the additional edge $(2, 1)$
- Graph L : nodes $2, 3, \dots, n$ are fully connected and there are the additional edges $(p, 1)$ for all nodes $p \in \{3, 4, \dots, n\}$

We easily check that both K and L are nonsplit. We consider two executions of \mathcal{A} : The first with the communication graph K in the first round and the initial configuration $x_1(0) = x_2(0) = 0$, and $x_p(0) = 1$ for $p \in \{3, 4, \dots, n\}$. The second with the communication graph L in the first round and the initial configuration $x'_1(0) = 1$, $x'_2(0) = 0$, and $x'_p(0) = 1$ for $p \in \{3, 4, \dots, n\}$. Because of the definition of an averaging algorithm, in the first execution with the communication graph K , we have

$$x_1(1) = 0 \quad \text{and} \quad x_2(1) = a \in [0, 1].$$

Likewise, in the second one with L , we have

$$x'_1(1) = 1 \quad \text{and} \quad x'_2(1) = b \in [0, 1].$$

From the fact that $K \sim_2 L$, $x_2(0) = x'_2(0)$, and $x_p(0) = x'_p(0)$ for all incoming neighbors p of process 2 in either of the two graphs, we deduce that $a = b$. Since \mathcal{A} is α -contracting, it follows that

$$a \leq \alpha \quad \text{and} \quad a \geq 1 - \alpha.$$

But this is only possible if $\alpha \geq 1/2$, which concludes the proof also in the case $n \geq 3$. \square

Both lower bounds in Theorem 6 are tight. Indeed the *midpoint algorithm* that we will introduce in Section 4.3 is $(1/2)$ -safe, and so $(1/2)$ -contracting in the network model of nonsplit communication graphs. For a two process network, Algorithm 1 has a contraction rate of $1/3$: we easily check that for every positive integer k ,

$$|x_1(k) - x_2(k)| = \frac{|x_1(k-1) - x_2(k-1)|}{3}$$

be the communication graph at round k equal to H^+ , H^- , or G .

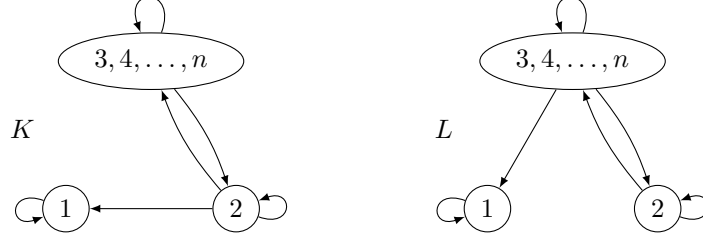


Figure 2: Communication graphs K and L used in the proof of Theorem 6 for the case $n \geq 3$

Algorithm 1 Averaging algorithm for two processes with contraction rate $1/3$

Initialization:

1: $x_p \in [0, 1]$

In round $k \geq 1$ do:

2: send x_p to other process and receive x_q if $(q, p) \in E(k)$

3: **if** x_q was received **then**

4: $x_p \leftarrow x_p/3 + 2x_q/3$

5: **end if**

4 Amortized averaging algorithms

Proposition 4 and Theorem 5 suggest to consider a new type of distributed algorithms, that we call *amortized averaging algorithms*, in which each process repeatedly first collects values during $n - 1$ rounds, and then computes a weighted average of values it has received every $n - 1$ rounds. Roughly speaking, an amortized averaging algorithm is an averaging algorithm with the granularity of macro-rounds consisting in blocks of $n - 1$ consecutive rounds.

We now make this notion more precise. First let us fix some notation. Macro-round ℓ is the sequence of rounds $(\ell - 1)(n - 1) + 1, \dots, \ell(n - 1)$. We consider algorithms for which every variable x_p is updated only at the end of macro-rounds; $x_p(\ell)$ will denote the value of x_p at the end of round $\ell(n - 1)$, as no confusion can arise. Given some communication pattern $(G(k))_{k \geq 1}$, the communication graph at macro-round ℓ is equal to:

$$\hat{G}(\ell) = G((\ell - 1)(n - 1) + 1) \circ \dots \circ G(\ell(n - 1)).$$

Each process p can record the set of values it has received during macro-round ℓ , namely the set $V_p(\ell) = \{x_q(\ell - 1) \mid q \in \text{In}_p(\hat{G}(\ell))\}$, but in anonymous networks, p cannot determine the set of its incoming neighbors in $\hat{G}(\ell)$. This is in contrast to networks with unique process identifiers where each process p can determine the membership of $\text{In}_p(\hat{G}(\ell))$ by piggybacking the name of sender onto every message, and so can compute the set $W_p(\ell) = \{(q, x_q(\ell - 1)) \mid q \in \text{In}_p(\hat{G}(\ell))\}$. In particular, each process can determine the multiset of values that it has received during a macro-round, counted with their multiplicities.

In consequence in any anonymous network, we can define the *amortized version* of an averaging algorithm \mathcal{A} with weights in update rules that depend only on the sets of received values: at the end of every macro-round ℓ , each process p adopts a new value by applying the same update rule as in \mathcal{A} with the macro-set $V_p(\ell)$. Based on the above discussion, this definition can be extended to averaging algorithms with update rules involving the sets of incoming neighbors when processes have unique identifiers. For instance, the amortized version of the mean-value algorithm is defined in any anonymous network while the amortized equal-neighbor algorithm requires to have unique process identifiers.

In both cases, the new value adopted by process p at the end of macro-round ℓ lies within the interval formed by the values of its incoming neighbors in the communication graph $\hat{G}(\ell)$: if $\hat{m}_p(\ell - 1)$ is the minimum of the values in $V_p(\ell)$ and $\hat{M}_p(\ell - 1)$ is the maximum, then

$$\hat{m}_p(\ell - 1) \leq x_p(\ell) \leq \hat{M}_p(\ell - 1).$$

Combining Proposition 4 and Theorem 5, we immediately derive the following central result for amortized averaging algorithms.

THEOREM 7. *In any rooted network model, the amortized version of a ϱ -safe averaging algorithm solves approximate consensus and achieves ε -agreement in $(n-1)\lceil \log_{\frac{1}{1-\varrho}} \left(\frac{1}{\varepsilon}\right) \rceil$ rounds.*

In order to fix decision times, we have assumed from the beginning that each process knows the number n of processes or at least an upper bound on n . However, regarding the *asymptotic consensus* problem – obtained by substituting limit values for decision values in the specification of approximate consensus –, this global knowledge on n is actually useless for averaging algorithms.

In contrast, update rules in amortized averaging algorithms require that the number of processes in the network is known to all processes. In other words, amortized averaging algorithms are defined only under the assumption of this global knowledge in the network, even for solving asymptotic consensus. In fact, we can immediately adapt the definition of amortized averaging algorithms to the case where n is a fixed parameter and then easily verify that Theorem 7 still holds when n is only an upper bound on the exact number of processes.

4.1 A quadratic-time algorithm in rooted networks with process identifiers

In [5], we proposed an approximate consensus algorithm with a quadratic decision time. The algorithm (cf. Algorithm 2) does not work in anonymous networks as it uses process identifiers so that each process can determine whether some value that it has received several times during a macro-round is originated from the same process or not.

Algorithm 2 Amortized equal-neighbor algorithm

Initialization:

1: $x_p \in [0, 1]$ and $W_p \leftarrow \{(p, x_p)\}$

In round $k \geq 1$ do:

2: send W_p to all processes in $\text{Out}_p(k)$ and receive W_q from all processes q in $\text{In}_p(k)$

3: $W_p \leftarrow \bigcup_{q \in \text{In}_p(k)} W_q$

4: **if** $k \equiv 0 \pmod{n-1}$ **then**

5: $x_p \leftarrow \frac{1}{|W_p|} \sum_{(q,v) \in W_p} v$

6: $W_p \leftarrow \{(p, x_p)\}$

7: **end if**

The update rule (line 5) rewrites as:

$$x_p(\ell) = \frac{1}{|\text{In}_p(\hat{G}(\ell))|} \sum_{q \in \text{In}_p(\hat{G}(\ell))} x_q(\ell-1).$$

Hence Algorithm 2 is the amortized version of the equal-neighbor algorithm.

From Proposition 2 and Theorem 7, it follows that Algorithm 2 solves approximate consensus. Moreover, because of the inequality $\log(1-a) \leq -a$ when $0 \leq a$ and because $\delta(x(0)) \leq 1$, it follows that if $\ell \geq n \log \frac{1}{\varepsilon}$, then $\delta(x(\ell)) \leq \varepsilon$. Hence Algorithm 2 achieves ε -agreement in $O(n^2 \log \frac{1}{\varepsilon})$ rounds.

4.2 A quadratic-time algorithm in anonymous rooted networks

As we will show below, Algorithm 2 still works when processes just collect values without taking into account processes from which they originate.

At each macro-round ℓ , the update rule in the resulting algorithm for anonymous networks (cf. Algorithm 3, line 6) coincides with the update rule in the mean-value algorithm. In other words, Algorithm 3 is the amortized version of the mean-value algorithm. Since the latter algorithm is a $(1/n)$ -safe averaging algorithm (Proposition 2), we may apply Theorem 7 and obtain the following result.

THEOREM 8. *In a rooted network model of n processes, the amortized mean-value algorithm solves approximate consensus and achieves ε -agreement in $O(n^2 \log \frac{1}{\varepsilon})$ rounds.*

Algorithm 3 Amortized mean-value algorithm

Initialization:

- 1: $x_p \in [0, 1]$
- 2: $V_p \subseteq V$, initially \emptyset

In round $k \geq 1$ do:

- 3: send V_p to all processes in $\text{Out}_p(k)$ and receive V_q from all processes q in $\text{In}_p(k)$
 - 4: $V_p \leftarrow \bigcup_{q \in \text{In}_p(k)} V_q$
 - 5: **if** $k \equiv 0 \pmod{n-1}$ **then**
 - 6: $x_p \leftarrow \frac{1}{|V_p|} \sum_{v \in V_p} v$
 - 7: $V_p \leftarrow \emptyset$
 - 8: **end if**
-

4.3 A linear-time algorithm in anonymous rooted networks

We improve the above quadratic upper bound on decision times with a linear amortized averaging algorithm which differs from our previous algorithm in the update rule: each process adopts the mid-point of the range of values it has received during a macro-round (cf. Algorithm 4).

Algorithm 4 Amortized mid-point algorithm

Initialization:

- 1: $x_p \in [0, 1]$
- 2: $m_p \in [0, 1]$, initially x_p
- 3: $M_p \in [0, 1]$, initially x_p

In round $k \geq 1$ do:

- 4: send (m_p, M_p) to all processes in $\text{Out}_p(k)$ and receive (m_q, M_q) from all processes q in $\text{In}_p(k)$
 - 5: $m_p \leftarrow \min \{m_q \mid q \in \text{In}_p(k)\}$
 - 6: $M_p \leftarrow \max \{M_q \mid q \in \text{In}_p(k)\}$
 - 7: **if** $k \equiv 0 \pmod{n-1}$ **then**
 - 8: $x_p \leftarrow (m_p + M_p)/2$
 - 9: $m_p \leftarrow x_p$
 - 10: $M_p \leftarrow x_p$
 - 11: **end if**
-

Let us now consider the *mid-point algorithm* that is the simple averaging algorithm with the mid-point update rule. In other words, Algorithm 4 is the amortized version of the mid-point algorithm.

By definition, the mid-point algorithm is $(1/2)$ -safe. By Theorem 5, it follows that the mid-point algorithm is a $(1/2)$ -contracting approximate consensus algorithm in any nonsplit network model, with a constant decision time. This improves the linear time complexity of the equal neighbor averaging algorithm [5] for this type of network models and demonstrates that the $1/2$ lower bound in Theorem 6 for networks with $n \geq 3$ processes is tight.

Furthermore Theorem 7 implies that Algorithm 4 solves approximate consensus in any rooted network model with a decision time in $O(n \log \frac{1}{\varepsilon})$ rounds.

THEOREM 9. *In a rooted network model of n processes, the amortized mid-point algorithm solves approximate consensus and achieves ε -agreement in $(n-1) \lceil \log_2 \frac{1}{\varepsilon} \rceil$ rounds.*

Hence the amortized mean-value algorithm and the amortized mid-point algorithm both work in any anonymous network model, under the assumption that the number of processes is known to all processes. However while the first algorithm has quadratic decision times and requires each process to store n values per round and to transmit n values per message, the amortized mid-point algorithm solves approximate consensus in linear-time and with only a constant number of (namely, two) values per process and per message. A similar result has been recently obtained by Olshevsky [18] with a linear-time algorithm where each process maintains two variables. Unfortunately, this algorithm works with a fixed communication graph that further ought to be bidirectional and connected.

5 Resiliency of amortized averaging algorithms

In this section, we discuss the resiliency of our amortized averaging algorithms against a wrong estimate of the number of processes or against a partial failure of the assumption that communication graphs are permanently rooted.

Firstly consider some communication pattern in which only part of communication graphs are rooted: suppose that $N-1$ communication graphs in any macro-round of $n-1$ consecutive rounds are guaranteed to be rooted. Then there are at least $n-1$ rooted communication graphs in any sequence of $L = \left\lceil \frac{n-1}{N-1} \right\rceil$ macro-rounds of length $n-1$, and every product of L communication graphs of macro-rounds is nonsplit.

Secondly suppose that the network model is indeed rooted but processes do not know the exact number n of processes and only knows an estimate N on n . Macro-rounds in the amortized averaging algorithms then consist of $N-1$ rounds (instead of $n-1$), and so the cumulative communication graphs in such macro-rounds may be not nonsplit when $N < n$. However the communication graph over any block of $L = \left\lceil \frac{n-1}{N-1} \right\rceil$ macro-rounds of length $N-1$ is nonsplit.

In both cases, the above discussion leads to introduce the notion of K -nonsplit network model defined as any network model \mathcal{N} such that every product of K communication graphs from \mathcal{N} is nonsplit. Theorem 5 can then be extended as follows:

THEOREM 10. *In a K -nonsplit network model, a ϱ -safe averaging algorithm is $(1 - \varrho^K)$ -contracting over each block of K consecutive rounds, i.e., for every non negative integer k , we have*

$$\delta(x(k+K)) \leq (1 - \varrho^K) \delta(x(k)).$$

Proof. Let p and q two processes such that $\delta(x(k+K)) = x_p(k+K) - x_q(k+K)$. Because the communication graph $G(k+1) \circ \dots \circ G(k+K)$ is nonsplit, there exist $2K+1$ processes $p_1, \dots, p_K, q_1, \dots, q_K, r$ such that

$$\begin{cases} p_K = p, q_K = q, \\ p_{i-1} \in \text{In}_{p_i}(G(k+i)), q_{i-1} \in \text{In}_{q_i}(G(k+i)), \text{ for each integer } i, 2 \leq i \leq K \\ r \in \text{In}_{p_1}(G(k+1)) \cap \text{In}_{q_1}(G(k+1)). \end{cases}$$

Let p_K^+ and p_K^- denote two processes such that

$$x_{p_K^+}(k+K-1) = M_{p_K}(k+K-1) \text{ and } x_{p_K^-}(k+K-1) = m_{p_K}(k+K-1).$$

Similarly we introduce two processes q_K^+ and q_K^- . Because the algorithm is ϱ -safe, we have

$$\begin{aligned} x_{p_K}(k+K) - x_{q_K}(k+K) &\leq (1 - \varrho)(x_{p_K^+}(k+K-1) - x_{q_K^-}(k+K-1)) \\ &\quad + \varrho(x_{p_K^-}(k+K-1) - x_{q_K^+}(k+K-1)) \\ &\leq (1 - \varrho)(M(k) - m(k)) + \varrho(x_{p_{K-1}}(k+K-1) - x_{q_{K-1}}(k+K-1)). \end{aligned}$$

By iterating the same argument K times, we obtain

$$x_{p_K}(k+K) - x_{q_K}(k+K) \leq (1 - \varrho)(1 + \varrho + \dots + \varrho^{K-1})(M(k) - m(k)) + \varrho^K(x_r(k) - x_r(k)).$$

Hence we have

$$\delta(x(k+K)) \leq (1 - \varrho^K) \delta(x(k)),$$

which concludes the proof. \square

As an immediate corollary of Theorem 10, we obtain that the amortized version of a ϱ -safe averaging algorithm is resilient against a wrong estimate of the number of processes or against a partial failure of the assumption of a rooted network model, and its decision times are multiplied by a factor in $O(L\varrho^{-L})$. Note that the theorem measures the number of macro-rounds, and not the number of rounds.

THEOREM 11. *The amortized version of a ϱ -safe averaging algorithm solves approximate consensus even with an erroneous number of processes or with a communication pattern where only part of communication graphs are rooted. Moreover it achieves ε -agreement in $O(L\varrho^{-L} \cdot \log(\frac{1}{\varepsilon}))$ macro-rounds if N denotes the estimate on process number or if only $N-1$ rounds in each block of $n-1$ consecutive rounds are guaranteed to have a rooted communication graph and where $L = \left\lceil \frac{n-1}{N-1} \right\rceil$.*

6 Rounding

In this section, we take into account the additional constraint that processes can only store and transmit quantized values. This model provides a good approximation for networks with storage constraints or with finite bandwidth channels.

We include this constraint by quantizing each averaging update rule. For that, we fix some positive integer Q and choose a rounding function, denoted $\lfloor \cdot \rfloor$, which rounds down (or rounds up) to the nearest multiple of $1/Q$. Then the quantized update rule for process p at round k writes

$$x_p(k) = \left\lfloor \sum_{q \in \text{In}_p(k)} w_{qp}(k) x_q(k-1) \right\rfloor$$

where the $w_{qp}(k)$ denote the weights in the average of the original algorithm. Besides we assume that all initial values are multiples of $1/Q$.

6.1 Quantization and mid-point

Nedić et al. [17] proved that in any strongly connected network model, every quantized averaging algorithm with the update rule (6) solve exact consensus (and so approximate consensus). Because of the impossibility result for exact consensus [5], their result does not hold if the strong connectivity assumption is weakened into the one of rooted network models.

For the same reason, if $\varepsilon < 1/Q$, then ε -consensus cannot be generally achieved in a rooted network model by a quantized averaging algorithm or its amortized version. In this section, we prove that the quantization of the amortized mid-point algorithm indeed achieves $1/Q$ -agreement in any rooted network model. In case one is interested in obtaining a (suboptimal) precision $\varepsilon > 2/Q$, a small adaptation of the proof yields that this is already achieved earlier on in round $(n-1) \left(\lfloor \log_2 \frac{Q-2}{Q\varepsilon-2} \rfloor + 1 \right)$.

THEOREM 12. *In a rooted network model, quantization of the amortized mid-point algorithm achieves $1/Q$ -agreement by round $(n-1) \left(\lfloor \log_2(Q-2) \rfloor + 2 \right)$. Moreover in every execution, the sequence of values of every process p converges to a limit x_p^* that is a multiple of $1/Q$ in finite time, and for every pair of processes p, q , we have either $x_p^* = x_q^*$ or $|x_p^* - x_q^*| = 1/Q$.*

Proof. We adopt the same notation as in the previous sections; in particular, we set $m(\ell) = \min_{p \in [n]} (x_p(\ell))$ and $M(\ell) = \max_{p \in [n]} (x_p(\ell))$. Moreover we give the proof in the case of the rounding down rule.

Since the mid-point algorithm is $(1/2)$ -safe in any nonsplit network model, for every macro-round ℓ we have

$$\delta(x(\ell)) \leq \frac{\delta(x(\ell-1))}{2} + \frac{1}{Q}.$$

Hence we obtain

$$\delta(x(\ell)) \leq \frac{\delta(x(0))}{2^\ell} + \frac{1}{Q} \left(1 + \frac{1}{2} + \dots + \frac{1}{2^{\ell-1}} \right) \leq \frac{1}{2^\ell} + \frac{2}{Q} \left(1 - \frac{1}{2^\ell} \right). \quad (3)$$

It follows that $\delta(x(\ell)) < \frac{3}{Q}$ for every macro-round ℓ with $\ell > \log_2(Q-2)$.

Let $L = \lfloor \log_2(Q-2) \rfloor + 1$; we have $\delta(x(L)) < \frac{3}{Q}$, and so at the end of macro-round L , processes adopt at most three different values which are multiples of $1/Q$ in an interval of the form $[(j-1)/Q, (j+1)/Q]$. We now consider the following three cases:

1. $M(L) = m(L)$; then all the x_p 's remains equal to $M(L)$ from macro-round L . The theorem trivially holds in this case.
2. $M(L) = m(L) + 1/Q$. By an inductive argument, we see that for each process p , the sequence $(x_p(\ell))_{\ell \geq 1}$ is eventually constant⁵ and its limit is either $m(L)$ or $m(L) + 1/Q$. This shows that the theorem holds in this case.

⁵Note that there is no bound on the convergence time.

3. Otherwise $M(L) = m(L) + 2/Q$. We show that $M(L+1) - m(L+1) \leq 1/Q$.

Let \mathcal{M} denote the set of processes with values $M(L)$ at the end of macro-round L . Then for every process p , we have

$$x_p(L+1) = \begin{cases} m(L) \text{ or } m(L) + 1/Q & \text{if } p \notin \mathcal{M} \\ m(L) + 1/Q \text{ or } m(L) + 2/Q & \text{if } p \in \mathcal{M} \end{cases}$$

Suppose for contradiction that $M(L+1) = m(L+1) + 2/Q$. Then we have $m(L+1) = m(L)$, $M(L+1) = M(L) = m(L) + 2/Q$, and there exists some process $q_0 \in \mathcal{M}$ such that

$$x_{q_0}(L+1) = x_{q_0}(L) = m(L) + 2/Q.$$

In the communication graph $\hat{G}(L+1)$ of macro-round $L+1$, all of q_0 's incoming neighbors are in \mathcal{M} . Combined with the fact that $\hat{G}(L+1)$ is nonsplit, this implies that every process p has an incoming neighbor in \mathcal{M} . Hence for every process $p \notin \mathcal{M}$, we have

$$x_p(L+1) = m(L) + 1/Q,$$

leading to a contraction with the equality $m(L+1) = m(L)$.

It follows that either $M(L+1) = m(L+1)$ or $M(L+1) = m(L+1) + 1/Q$. In other words, either case (1) or case (2) occurs at the end of macro-round $L+1$. \square

Inequality (3) can be easily generalized for the amortized version of any ϱ -safe algorithm. However the rest of the above proof works only for value $1/2$ of parameter ϱ . Hence Theorem 12 seems to be specific to the amortized mid-point algorithm.

6.2 Approximate consensus versus 2-set consensus

We now consider the *2-set consensus problem* which is another natural generalization of the consensus problem. Instead of requiring that processes agree to within any positive real-valued tolerance ε , processes have to decide on at most 2 different values:

Agreement. There are at most two different decision values.

Formally, each process starts with an input value from the set V of multiples of $1/Q$ and has to output a decision value from V in such a way that the termination and validity conditions in the consensus specification as well as the above agreement condition are satisfied.

The 2-set consensus problem naturally reduces to approximate consensus: processes round off their $1/Q$ -agreement output values. Unfortunately, the use of averaging procedures to solve approximate consensus leads processes to exchange values out of the set V in the resulting 2-set consensus algorithms. The quantized amortized mid-point algorithm allows us to overcome this problem, and Theorem 12 shows that in any rooted network model, this algorithm achieves 2-set consensus in $(n-1)(\lceil \log_2(Q-2) \rceil + 2)$ rounds.

The above discussion shows that the 2-set consensus problem is solvable in a dynamic network model if all the communication graphs are rooted. In particular, 2-set consensus is solvable in an asynchronous complete network with a minority of faulty senders since nonsplit rounds can be implemented in this model. Combined with the impossibility result in [10] in the case of a strict majority of faulty processes, we obtain an exact characterization of the sender faulty models for which 2-set consensus is solvable in asynchronous systems if the number of processes n is odd and a small gap (namely $n/2$ faulty processes) when n is even.

Our positive result can be interestingly compared with the 2 faulty processes boundary [2, 14, 20] between possibility and impossibility of the original (and stronger) 2-set consensus problem [7] where decision values ought to be initial values, instead of being in the range of the initial values. That points out the crucial role of the validity condition on the solvability of 2-set consensus.

References

- [1] Vincent D. Blondel, Julien M. Hendrickx, Alex Olshevsky, and John N. Tsitsiklis. Convergence in multiagent coordination, consensus, and flocking. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC)*, pages 2996–3000. IEEE, New York, NY, 2005.
- [2] Elizabeth Borowsky and Eli Gafni. Generalized FLP impossibility result for t -resilient asynchronous computations. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 91–100. ACM Press, 1993.
- [3] Ming Cao, A. Stephen Morse, and Brian D. O. Anderson. Reaching a consensus in a dynamically changing environment: a graphical approach. *SIAM Journal on Control and Optimization*, 47(2):575–600, 2008.
- [4] Bernadette Charron-Bost. Orientation and connectivity based criteria for asymptotic consensus. arXiv:1303.2043v1 [cs.DC], 2013.
- [5] Bernadette Charron-Bost, Matthias Függer, and Thomas Nowak. Approximate consensus in highly dynamic networks: The role of averaging algorithms. In *Proceedings of ICALP*, pages 528–539, 2015.
- [6] Bernadette Charron-Bost and André Schiper. The Heard-Of model: computing in distributed systems with benign faults. *Distrib. Comput.*, 22(1):49–71, 2009.
- [7] Soma Chaudhuri. More choices allow more faults: Set consensus problems in totally asynchronous systems. *Information and Computation*, 105(1):132–158, 1993.
- [8] Bernard Chazelle. The convergence of bird flocking. arXiv:0905.4241 [cs.CG], 2009.
- [9] Bernard Chazelle. The total s -energy of a multiagent system. *SIAM Journal on Control and Optimization*, 49(4):1680–1706, 2011.
- [10] Roberto De Prisco, Dahlia Malkhi, and Michael K. Reiter. On k -set consensus problems in asynchronous systems. In *18th ACM Symposium on Principles of Distributed Computing (PODC 1999)*, pages 257–265. ACM, New York City, 1999.
- [11] Jean-Charles Delvenne, Ruggero Carli, and Sandro Zampieri. Optimal strategies in the average consensus problem. *Systems & Control Letters*, 58(10–11):759–765, 2009.
- [12] R. Hegselmann and U. Krause. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3):1–33, 2002.
- [13] Julien M. Hendrickx, Raphaël M. Jungers, Alexander Olshevsky, and Guillaume Vankeerberghen. Graph diameter, eigenvalues, and minimum-time consensus. *Automatica*, 50(2):635–640, 2014.
- [14] Maurice Herlihy and Nir Shavit. The asynchronous computability theorem for t -resilient tasks. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 111–120. ACM Press, 1993.
- [15] Renato E. Mirollo and Steven H. Strogatz. Synchronization of pulse-coupled biological oscillators. *SIAM Journal on Applied Mathematics*, 50(6):1645–1662, December 1990.
- [16] Luc Moreau. Stability of multiagent systems with time-dependent communication links. *IEEE Transactions on Automatic Control*, 50(2):169–182, 2005.
- [17] Angelia Nedic, Alexander Olshevsky, Asuman E. Ozdaglar, and John N. Tsitsiklis. On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control*, 54(11):2506–2517, 2009.
- [18] Alex Olshevsky. Linear time average consensus on fixed graphs. *IFAC-PapersOnLine*, 48(22):94–99, 2015.

- [19] Alex Olshevsky and John N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Review*, 53(4):747–772, 2011.
- [20] Michael E. Saks and Fotios Zaharoglou. Wait-free k -set agreement is impossible: the topology of public knowledge. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 101–110. ACM Press, 1993.
- [21] Nicola Santoro and Peter Widmayer. Time is not a healer. In B. Monien and R. Cori, editors, *6th Symposium on Theoretical Aspects of Computer Science*, volume 349 of *LNCS*, pages 304–313. Springer, Heidelberg, 1989.
- [22] Steven H. Strogatz. From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators. *Physica D: Nonlinear Phenomena*, 143(1–4):1–20, 2000.
- [23] Y. Yuan, G.-B. Stan, L. Shi, M. Barahona, and J. Goncalves. Decentralized minimum-time consensus. *Automatica*, 49(5):1227–1235, 2013.